Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Haseeb Shehzad

Time Series Analysis and Prediction:

Service Request data from real estate

Master's Thesis
Espoo, October 22, 2019

August 23, 2020

Supervisors:	Assistant Professor Pekka Martinen, Aalto University
Advisor:	Ossi Talvitie, Fatman Oy

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Haseeb Shehzad		
Time Series Analysis and Prediction:			
Time Series Analysis and Prediction: Service Request data from real estate			
Date:	October 22, 2019	**Pages:**	48
Major:	Machine Learning and Data Mining	**Code:**	SCI3042
Supervisors:	Assistant Professor Pekka Martinen		
Advisor:	Ossi Talvitie, Fatman Oy		

Time series data is generated by various domains, fields and industries and has found many applications with the increase in the availability of different type of data. In this thesis work, we analyzed the time series data of service requests from real-estates and captured the yearly seasonality and trend from the past data and forecast the number of service requests for incoming months. We present two approaches to achieve our goal, structural time series and ARIMA models. We concluded that the structural time series performed better for predicting the future values of the data, although the seasonality and trend components from the past data were almost the same from both models.

Keywords:	time, series, analysis, prediction, machine learning, statistics
Language:	English

Acknowledgements

I would like to thank the following people without whom I would not have been able to complete my thesis and without whom I would not have made it through my masters degree:

My thesis supervisor, Mr. Pekka Martinen for providing guidance and feedback. Fatman Oy for the opportunity to work on this project and supporting me throughout this project and my degree. Dear friends and family. To my parents, who set me off on the road to this MSc a long time ago.

Espoo, October 22, 2019

Haseeb Shehzad

Abbreviations and Acronyms

KPSS	Kwiatkowski Phillips Schmidt Shin
ADF	Augmented Dickey Fuller
MCMC	Markov chain Monte Carlo
MAP	Mean average precision
AIC	Akaike information criterion
BIC	Bayesian information criterion
MSE	Mean squared error
MAPE	Mean absolute percentage error
RMSE	Root mean square error
MAE	Mean absolute error
AR	Autoregressive
MA	moving average
ARMA	Autoregressive moving average
ARIMA	Autoregressive integrated moving average
ERP	Enterprise resource planning

Contents

List of Tables

Chapter 1

Introduction

Time series data is a series of events measuring some quantity over time. Most often it could also be described as equally spaced data points in time. There are many processes that are generating the time series data in real-time and the availability of such data is growing exponentially. The time series data could be generated by a natural process for example the climate change in terms of weather and other factors, number of earthquakes each year and spread of an epidemic over time. Or it could be human generated process, few examples could be the financial data of stock exchanges changing over time, consumption of resources e.g. electricity, water and gas over time and number of flights recorded each month in an airport. In the field of social sciences, there are population series e.g. birthrates, number of enrolments in schools etc. In medicine, we have blood sugar levels traced over time which could be helpful in understanding the diabetes patient's routines. With the growing availability of time series data, there is huge potential that this data contains the information about the processes that are generating this data and we could learn this process and predict the data of a business or social interest. The business process information would enable us to make better decisions based on predictions generated by different models. Also the past trend from the data would help the stakeholders to understand the overall performance of a facility.

We would implement and compare different forecasting and time series analysis techniques, methods and algorithms to take advantage of the opportunities present in the past data of Service Requests in Real estate. We would start with basic properties of the time series data, then present a statistical analysis of the time series and finally try to predict the data. We would use different state-of-the-art techniques and methods as well as classical and traditional methods to achieve this. Our objective in time series analysis is to train models that will provide the characteristics and data generation process of the real estate data.

1.1 Motivation

It is reported that 50% of the real estate management involve maintenance and rehabilitation in housing sector and 43% in non-residential areas in EU. [21] The cost can occur at three stages of a building's service life of 50 years; design, construction and maintenance or usage. According to Perret [17], about 75–80% cost occur during maintenance and usage stage. Large real estate companies spent enormous amount of money on maintenance and service requests. Thus, it is very important for these organizations to engage in capacity and resource planning. Forecasting the possible future cost of real estate maintenance can result in better customer satisfaction and decision making. As we will see in results section in detail that we are able to identify the busiest month for service requests from past data as annual trend to help the real estate owners and leasers to plan in advance for these months. This serves the dual purpose; they can figure out the rent prices based on future maintenance cost of the facility and they can provide better service for their tenants.

Occupancy costs can hurt a company's income, share value, and growth. Also the real estate cost is usually the second biggest cost after the payroll cost in most of the organizations [24] and 20–40% corporate value. So, it will also help the real estate lessee to figure out the future maintenance cost of the facility and they might seek another option or negotiate the rent prices.

1.2 Service Requests from Real estate

There are advance Enterprise Resource Planning (ERP) systems that enable tenants to create a service request and notify property owners or subcontractors about a problem or malfunction in the facility. These ERP systems save information about the life cycle (start, progress, completed etc.) of a service request job. There could be many sources of a service request; it could be made during an inspection of the building when an inspector finds a defect in the facility, or a maintenance plan might require a service request to be created, or it could be the finding of a tenant or an internal personnel. As soon as a service request is created, it is assigned to the service company personnel responsible for that specific building and a specific fault type of problem in the facility. Service request could be of any fault type (electricity, water, cleaning, heat etc.) and it is specified while creating it. There could be statuses of service requests showing in what state it is at that time to keep track of all the service requests. An example of the life cycle of an average service request could be described as followed, a newly created service request is in 'received' status, once a service company personnel start working on the task, it will be changed to 'in progress' status and when it is completed, it will be in 'closed' status.

There is a difference between planned long or short-term maintenance and service requests. Service requests are made when there is a fix required of an urgent nature and it is something small enough that can be fixed within few days. Also, service requests are created by tenants while maintenance tasks are planned by building owners as a requirement of facility planning and maintenance laws. We have gathered data from Frame ERP software from a real estate owner and a user of Frame. We are dealing this data as time series data as the creation of service requests is considered to be equally time spaced since we are taking the monthly count. Considering this as Time series gives us the advantage of finding patterns, trends and see the effect of seasons and time of the year.

We will try to forecast and analyse the occurrence of service requests (monthly frequency). In addition to these predictions, we will get useful statistics from data. We will try to find patterns, trends and seasonality in data using different Time Series analysis methods.

1.3 Research Question and Purpose

We are interested in knowing that are we able to predict and analyse the past time series data of service requests for real estate. We assume that there are business cycle, seasonal and trend components in the data that could be learned and predicted using the different time series methods and models.

1.4 Structure of the Thesis

The thesis starts with the introduction to the general overview of time series and followed by the motivation and need behind this thesis work. The use case and the where the data originated from was explained in the next section. The background of the used methods are further discussed as well as some important topics and terms of time series. The structural time series and Arima models are described next along with the trend and seasonality components of both models. The data source, pre-processing of data and the nature of data under-study was discussed before moving forward with the discussion about how these two discussed models are implemented and what results were obtained from these experiments. We discussed the evaluation process used for both models and discussed the findings from the experiments and the thesis work. Finally, the thesis is concluded by discussing the achieved goal, future work directions and required improvements in the implementation.

Chapter 2

Background

2.1 Business Life Cycle in Real Estate

The business cycles are very important aspect of the modern economics that help policy makers to understand the fluctuations and growth over time. This helps them to improve the future policies and decisions based on past data and predictions. We will discuss these fluctuations as cycle and growth as trend in Section 2.3 in more detail next. We will use different analysis and prediction models to extract and forecast the business cycle, trend and seasonality from the past data of real estate. The data is gathered from the Frame ERP system for real estate. The Frame ERP system helps facility owners to maintain their facilities using the advance tools and functionalities available in the product. The data describes how many service requests were created each month over the period of almost 10 years for different facilities. We will try to understand the overall trend, available cycles and personalities to help the facility owners or service providers to engage in capacity planning and provide better customer service.

2.2 Time Series

Time series is a set of observed data points recorded over time. The discrete time series (the type this thesis in devoted to) is a set of observed variables that were made at fixed intervals. While in continuous time series, the data is gathered continuously with some time interval [4]. The graph 2.1 depicts a univariate time series under study in this thesis work. We will make assumptions that this time series is the realization of a stochastic process [8] and we will construct linear models to approximate the process that is generating this data. We would also assume that there are some regularities in the data and we will try to explain them by using the monthly frequency of service requests as a single available variable.

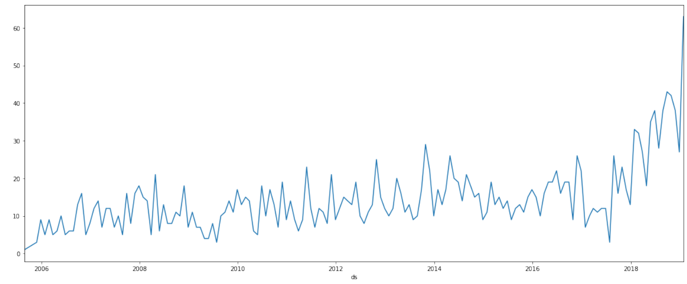

Figure 2.1: The number of service requests created in a month

2.3 Characteristics of Time Series

We can define time series as a sequence of random points $x_1, x_2, x_3, ...$, where x_1 denotes the value at first time point t_1, x_2 denotes the value at time point t_2, x_3 denote the value at time t_3 and so on. The time series described as random variables x_t indexed based on the order they were recorded or observed in time t is called stochastic process. The values that we have observed in a stochastic process are called realizations of the stochastic process.

We conventionally draw a time series data points on y-axis and time scale on x-axis. It is very convenient to draw these random points as adjacent points in a continuous manner in graphs visually even these points might have generated discretely in a time series. The series in many some cases appear as continuous smooth graph. We can explain this smoothness by the supposition that the data point x_t at t depends on past data point $x_t{-}1, x_t{-}2, ...$ meaning that these point are correlated. In our use case, we have univariate time series which involves the measurement of same variable (number of service request per month) over time. We will now discuss some important characteristics of time series and try to describe the time series data for service requests in the light of same discussion.

2.3.1 Cycles

The cycles in the time series data can be seen in contrast with the seasonal fluctuations. The cyclic patterns in the data do not have fixed period which means there might be a period in which the cycle is not present while the seasonality patterns are repeated in every period. We can think of business cycles which could last for many years but we usually don't know the length of the cycle hence the period is not fixed. A series could have both cyclic and seasonal components. The autoregressive structure in the series produces the cycles in the data while the seasonal components is always an external influence as we will discuss in the next section on seasonality. The Figure 2.2 shows a long run cycle in the sales of a company which is different from the yearly seasonality.

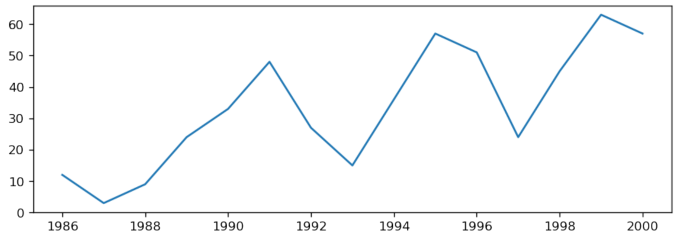

Figure 2.2: Sales of a company in millions of dollars

2.3.2 Seasonality

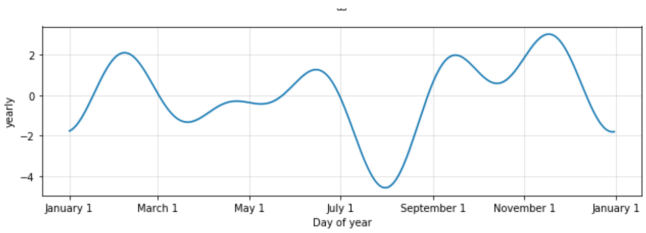

Figure 2.3: Month of the seasonality for service requests

Seasonality is the occurrence of some specific event or variation at regular intervals. It could be weekly, monthly, quarterly or yearly etc. repetition. Business time series usually have some seasonality effect in the data which could be explained by the human behaviour, weather conditions or vacations depending on the time series data generation process. For example, in our time series data from service requests, the seasonality graph 2.3 generated from the past data shows that the most number of service requests are created at the end of each year probably because people would like to get everything fixed before winter holidays. On the other hand, since the most number of people are on holidays around August and it explains the least number of service requests around August. This will be elaborated in discussions section in detail.

2.3.3 Trend

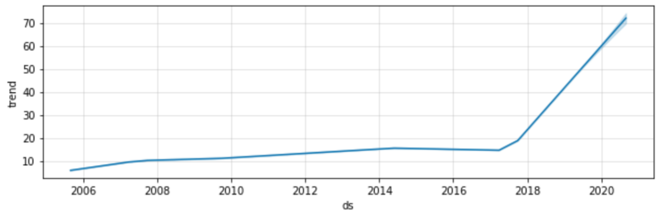

Figure 2.4: Seasonality effect of number of service requests creation

The third component in a time series is a trend. The trend actually shows how the series have grown and how it will continue to grow. In our example time series data trend shown in Figure 2.4, we noticed a constant growth in the data which is similar to the natural ecosystem. The growth is non-linear and saturates at the carrying capacity. The carrying capacity in our case is the number of tenants who can actually create service requests in a facility. This trend was noticed in almost every facility's series and could be explained by the fact that the real estate have more problem over time as the equipment and facility gets old and probably require more maintenance compared to a new facility. Another reason could be the number of tenants may have increased over time causing an increase in the number of service requests.

2.3.4 White Noise

A time series could be a series of uncorrelated data points with mean 0 and finite variance σ_w^2. By definition, it is not possible to model white noise time series and predict it. It is expected that the real world time series would have some white noise on top of the original signal. For the same reason, it could be used as a noise in many data science applications. The naming is taken from the analogy of white light and depicts that all the possible periodic signals have the same strength. Gaussian white noise could be useful noise with independent and identical random variables with mean 0 and variance σ_w^2 or $w_t \sim iid(0, \sigma_w^2)$.

Another important point to mention is the model diagnostics. The error series from a forecast model of time series should ideally be a white noise. When the forecast error is white, it shows that we were able to capture all the information in the signal and the rest is just a collection of random variables that could not be modelled and predicted.

2.3.5 Moving Averages

Moving average is an indicator of the time series that smooth out the short-term trends and highlights the long-term trends and cycles. It can be calculated by creating a timeline of averages of different subsets of the whole data. The mean function could be defined as

$$\mu_{xt} = E(x_t) = \int_{-\infty}^{\infty} x f_t(x) dx \tag{2.1}$$

where E denotes the expected value.

We can replace the w_t with a moving average in the white noise. We will replace the average of current and its previous values in the past and future,

$$v_t = \frac{1}{3}(w_t - 1 + w_t + w_t + 1) \tag{2.2}$$

which will show the smoother version of the white noise time series by highlighting the slower oscillations (long-term) and fading out the faster oscillations (short-term).

2.3.6 Stationarity

The regularity of a time series could be described in terms of stationarity. In strictly stationary time series, the probabilistic behaviour of all the subset of the data is identical to the one which is time shifted. The multivariate distribution of the subsets of variables of strictly stationary time series must agree with the time shifted set of all data points. While in weakly stationery time series x_t, the mean μ_t defined in 2.1 is a constant value and it does not depend on the time t. The autocovariance function

$$\gamma_t(s, t) = cov(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)] \tag{2.3}$$

depends on the difference of s and t as $|s - t|$.

The regularity in mean and autocorrelation allows us to estimate these quantities by averaging. In general, it is important for time series to be stationary so that we are able to take average of the lagged values over time, and if the dependence structure is weak, it is very difficult to analyse the time series. But many real-world processes generate the non-stationary time series. One of the easy forms of non-stationary time series to work with is trend stationary. In which stationarity exists around a trend.

$$x_t = \mu_t + y_t \tag{2.4}$$

2.4 Univariate Time Series Models

2.4.1 ARIMA

Arthur Schuster(1851-1934) was the first to use the periodogram method to solve the sun-spot periodicity in 1906 [16]. This gave motivation to the find the periodicity in other natural phenomena using the same implementation. The disadvantages of this method was to not taking into account the external factors and supposition about the strict periodicity in the data. George Udny Yule (1871-1951) was the first to establish the autoregression model [28] to solve these problems and based on his work Evgeny Evgenievich Slutzky (1880-1948) created the moving summation model [22]. Yule developed the famous AR(2) model using harmonic functions for the pendulum and sun-spot data and similarly AR(4) model was established by assuming that the harmonic function has two periods. An English statistician Walker then extended Yule's idea and attained a more general AR(s) model [25]. Finally Herman Wold (1908-1992), a Swedish statistician proved the decomposition theorem and based on it, he attained the ARMA model [27]. ARMA model provided the framework for ARIMA model which was established in 1970 [3].

2.4.2 Structural Time Series Model

The structural time series is defined as [11]

> The principal univariate structural time series models are therefore nothing more than regression models in which the explanatory variables are functions of time and the parameters are time varying.

The advantage of this model is that it is interpretable immediately. Harvey and Todd (1983) showed strong arguments that the structural models perform better than ARIMA model after comparing the forecasts of both models. Harvey (1884) further showed that the structural models could be used to cycle microeconomics time series.

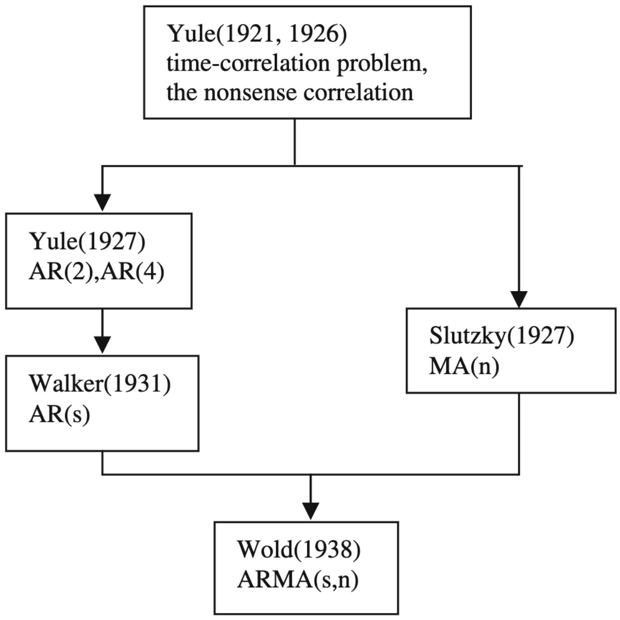

Figure 2.5: Development of time domain analysis methods [16]

Muth formulated the adaptive expectation as rational expectation in 1961, This expectation was represented by the linear stochastic and it was more general than the random walk and noise model. This representation became the major background for the structural time series. A structural time series (STM) is modelled by trend, seasonality and other disturbances in the data as unobserved components. STM is able to decompose the given time series into trend, seasonality, cycle and other irregular or noise components by estimating the maximum likelihood. This decomposition is called signal extraction. Each component has its own interpretation and it explains how the data varied because of this specific component.

We will consider a series to be sum of seasonal, trend and white noise components (4.1).

Chapter 3

Methods

3.1 Structural Time Series Model

Forecasting is a very important task in an organization in order to perform the capacity planning to allocate resources in timely manner and to measure how well the process is being performed in terms of achieving goals.

Usually a business forecasting requires deep domain knowledge of the business and underlying process. Sometimes the underlying process is so complex or unknown that it is a very big challenge to incorporate it in the time series modelling. Second challenge is the number of time series analysts compared to the variety of time series available from the point of view of the domain knowledge.

A modular regression model was proposed for time series analysts called Prophet. It includes the parameters that could be set and adjusted without the deep knowledge of the underlying methods.

It is thus important to tune these parameters effectively. This process is described in Figure 4.1.

When the prediction performance is not good, we adjust the parameters of the model according to the problem we are trying to solve and with the knowledge of the underlying time series models. We need, for example, the auto-regressive components, maximum orders of differencing and moving average components as parameters to the ARIMA model. The Prophet thus have the ability to be used as an automated method (with little knowledge of underlying time series models used and tuned visually) or by analysts-in-the-loop.

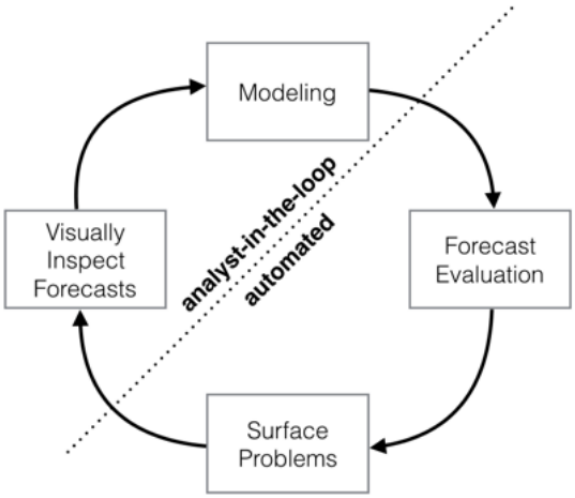

Figure 3.1: Systemic view of approach

3.1.1 Structural Time Series Model

The prophet uses a univariate, structural and decomposable model [10], which is decomposable into trend, seasonal and holidays components. One of the important aspects of the structural models is that it is formulated in terms of independent components, it consists of

$$y_t = g_t + s_t + h_t + \epsilon_t \qquad t = 1, ..., T \tag{3.1}$$

g_t is called the trend function and it could model the non-periodic changes (linear or logistic growth) in the the series data, s_t depicts the periodic changes which includes weekly, yearly or daily seasonality, while h_t represents the holidays effects which could occur on irregular basis on one or more days. Holidays could be see as shocks in the series and do not fall into the periodic pattern. ϵ_t is the error term and an irregular component which was not accommodated by the model.

The trend function g_t is a local approximation to the linear trend, i.e.

$$g_t = g_{t-1} + \beta_{t-1} + \eta_t \tag{3.2}$$

$$\beta_t = \beta_{t-1} + \zeta_t \qquad t = 1, ..., T \tag{3.3}$$

where η_t and ζ_t are independently distributed over time with mean 0 and variance σ_η^2 and σ_ζ^2 respectively. The process s_t that is generating the seasonal component could be described as,

$$s_t = -\Sigma_{j=1}^{y-1} s_{t-j} + \omega_t \qquad\qquad t = 1, ..., T \tag{3.4}$$

where ω_t is independently distributed error term with mean zero and variance σ_ω^2 while y is the number representing the seasons in the year. The seasonal pattern changes slowly over time but if we sum all of the seasonal components over y consecutive time periods, it would have a value of 0 and a constant variance. The equation 3.4 could be explained in terms of lag operator, L,

$$Y(L)s_t = (1 + L + ... + L^{y-1})s_t = \omega_t \qquad\qquad t = 1, ..., T \tag{3.5}$$

The disturbance or noise term ϵ_t is considered to be stationery unlike Trend and Seasonality terms with mean 0 and variance σ_ϵ^2.

3.1.2 Trend

We will discuss two, non-linear and linear trend models to cover application.

3.1.3 Non-linear Trend

The most important aspect of a growth model is that how the data generation process has created the data and how it is growing over time. We have noticed in the experiments that the service request data always keeps on growing just like a population growth model in nature maybe due to the fact the number of people renting the facilities have increased, the buildings and related equipment gets older and it require more maintenance over time. This growth could be modelled by logistic growth model,

$$g(t) = \frac{C}{1 + exp(-k(t - m))} \tag{3.6}$$

where c is called the carrying capacity, k describes the growth rate while m is called the offset parameter. The Figure 3.2 shows an example of non-linear trend (logistic growth).

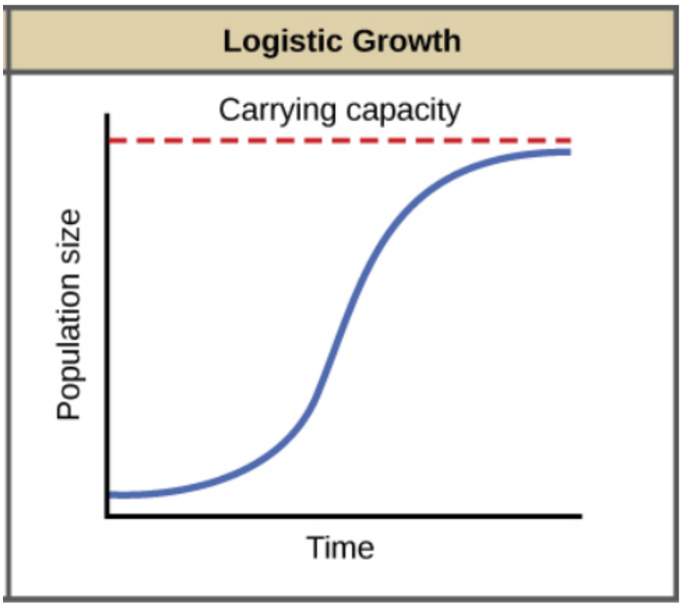

Figure 3.2: Logistic growth depicting the decrease in population when the carrying capacity of the environment is reached [1]

3.1.4 Linear Trend

A piece-wise linear trend could be defined as,

$$g(t) = (k + a(t)^T\delta)t + (m + a(t)^T\gamma) \tag{3.7}$$

where the k is called growth rate, δ is the rate adjustment, m is the offset parameter while γ_j could be set to $-s_j\gamma_j$ to achieve a continuous function. The s_j called the change point could be set by the analyst depending on the change in the growth rate of series or it could be selected automatically from a range of given change points by setting a prior on $\gamma_j \sim Laplace(0, \tau)$. The growth rate of the model can be controlled by the parameter τ thus defines the flexibility of the model. It is important to note that the above mentioned prior does not effect the growth rate k which means if we reduce the τ to zero, the model changes from logistic growth to linear growth. The Figure 3.3 show a linear trend fitted to global temperature deviation.

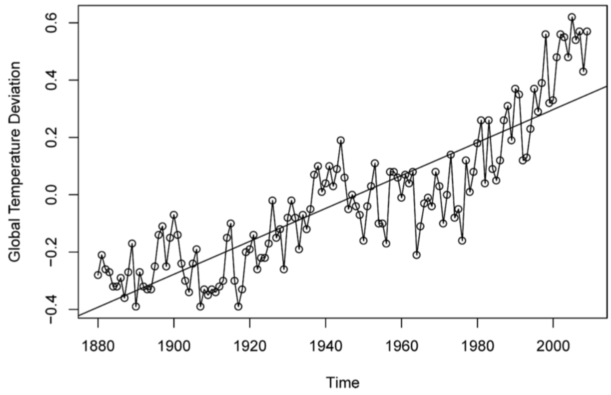

Figure 3.3: Global temperature deviation fitted with linear trend [20]

3.1.5 Seasonality

To identify and fit the seasonality components in a time series, we can rely on Fourier series e.g. see Figure A.3 for a flexible model [11]. If we assume P to be the regular period in a series while scaling the time variable in days, the P can take $P = 365.2$ for yearly and $P = 30$ for monthly data and so on. We used a standard Fourier series to approximate the smooth seasonal effects.

$$s(t) = \sum_{n=1}^{N} (a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})) \tag{3.8}$$

Since we are already fitting a trend term, we will leave out the intercept term. We will estimate 2N parameters, to achieve this we will form a matrix of seasonality vectors for each t from series data. In this case, $N = 160$ and $P = 30$, we can write 3.8 as

$$X(t) = [\cos(\frac{2\pi(1)t}{30}), ..., \sin(\frac{2\pi(160)t}{30})] \tag{3.9}$$

The seasonal component is given by

$$s(t) = X(t)\beta \tag{3.10}$$

where $\beta = [a_1, b_1, ..., a_N, b_N]^T$ are the parameters that could be taken from $\beta \sim Normal(0, \sigma^2)$ to smooth the prior of the seasonality.

3.2 ARIMA Model

Univariate time series are usually modelled by the autoregressive(AR) models or the Moving Averages models mentioned earlier in Section 2.3.5. AR model works like a linear regression by taking into account the assumption that the current value in time series depends on the weighted linear sum of the past values. So it could be analysed by linear least square in other methods by interpreting the ϕ_p parameter. We can formulate the AR as follows [14],

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \phi_p X_{t-p} + \epsilon_t \tag{3.11}$$

Where X_t is the observed time series, ϵ is white noise while p is the order of the AR. The AR would be infinite process if p approaches infinity. As we discussed in Section 2.3.5, MA could be described as the linear regression of one or more random shocks of the past series data. We can say that at each period t, a random shock is available which is independent of other shocks from all periods. Thus we can generate the time series by the weighted sum of all the shocks as follows [14],

$$X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} \tag{3.12}$$

where the q is the order of the MA process. Even though the MA is a linear regression process like AR, MA is more complicated compared to AR because it depends on the error terms from other periods which are not observable. To handle this we use an iterative and non-linear fitting and the estimation we get from it is not as interpretable as in AR models.

Jenkins and Box [3] first introduced the method of fitting and identifying both the AR and MA processes systematically which was investigated by Yule. ARMA model consisting of both of these components (weighted sum of past values and errors) is then defined as,

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} \tag{3.13}$$

Since both processes are stationary, the ARMA model is also assumed to be stationary given the condition for the roots of the polynomial $\phi(z) = 0$. We can further generate ARIMA(I for integrated) model if the ARMA model does not fulfil the stationarity conditions by taking the difference of the time series which will achieve the stationarity. Also we can add the seasonality components for both the AR and MA terms the same way we added for the non-seasonal components. Chatfield [5] Recommended to use at least 50 data points while 100 is mentioned by other authors too.

Chapter 4

Implementation

4.1 Data

4.1.1 Data Source

The time series data required for this thesis was collected from Frame ERP system for real estate. The Frame has a very popular feature in the facility management field that allows users and tenant to create, follow and give feedback about a maintenance problem in the facility. The Figure 4.1 shows a form used in Frame that provides users the ability to create, save, follow and most importantly notify the relevant service provider to get notified about this issue in the facility. The users of Frame can fill in all the required information and specify the receivers and send the form. All the sent service

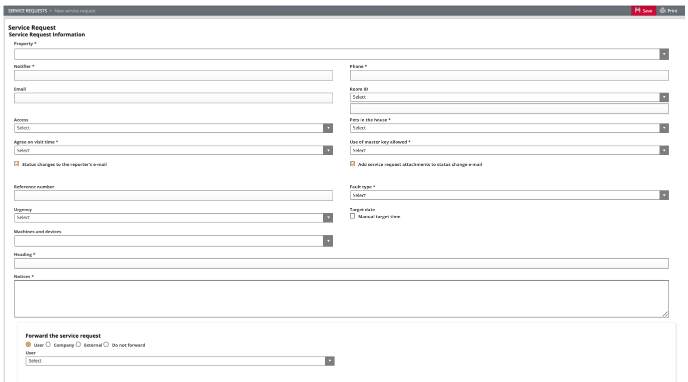

Figure 4.1: The service request creation form

requests forms are then available in the service providers account in Frame in a list view as shown in Figure 5.2 and responsible personnel can track and fix the issues and inform the users and tenants through the same interface. You can see different statuses of service requests shown in the Figure 4.2 that reflect the current situation of the request and make it easier for the users and administrators to follow the progress.

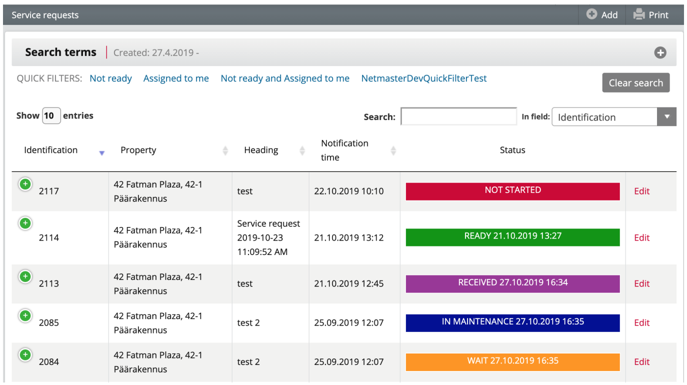

Figure 4.2: The list of received service requests

All of this data is available in the SQL database in a well structured form which could be easily used by any other system. We have selected a few important properties of the service request data and filtered the data based on these. For example, we have selected the data for each facility separately. It was important to deal the data for each facility separately for analysis and forecasting because each facility could have different factors affecting the number of service requests created, for example, the number of tenants in the facility, age of the building and service provider. The service requests are created on need basis and has a time stamp when it was generated so we have data in series of events making it obvious to deal it as a time series. The data was collected for a Frame customer which has a lot of facility around Finland that they sub rent to commercial users mostly. Most important fact that the customer has been using the software for a long time (more than 10 years) and they had all the data saved in the system. It was easier for us to capture the long run cycles from all this data and generally more data is useful in data analysis and forecasting in time series context for our methods also.

4.1.2 Data Collection and Preparation

We collected the data from our SQL database servers and selected a facility which had enough data as we run into a series of facilities which did not have enough data due to multiple reasons. The data was exported into excel files to be processed by the advance data analysis programming languages. This was just for the experiment purpose of this thesis. As a deliverable product, a data pipeline would be devised along with the front end graphics and prediction view but that's outside the scope of this thesis.

We converted the data from daily to monthly time series by simply taking the sum of all the created service requests by each month. The reason for this conversion was due to the fact that the service requests are not generated every day most of the time so it makes more sense to consider them on monthly level. It also makes sense from the use case point view also as we are mostly interested in long term changes (monthly trends and seasonality) and behaviour of the data and it does not help the administrators to know what is happening on daily basis as that is handled by other solutions in Frame. A simple graph of the data is shown in the Figure 4.4 and Figure 4.3 shows basic statistics of the data.

count	163.00
mean	14.99
std	8.36
min	3.00
25%	9.00
50%	13.00
75%	18.00
max	43.00

Figure 4.3: Basic statistics of service request data

4.1.3 Data Pre-processing

The data had very few outliers, which were removed to enhance the model performance and prediction. The removed outliers were found at the start of the time series because the number of service requests were very few in each month, the reason was supposed to be that the system was not in use that much at the start and as a result we didn't have much data. Also at the end of time series, the data had a huge number of service request and it was due to the general trend increase in the number of service requests. After doing some test in Section 4.1.4 to check the stationarity of time series, we concluded that the series is not stationary and we took the difference of consecutive terms in the data to achieve the stationarity shown in Section 3.2. This could be written as:

$$y_t = y_t - y_{t_1} \tag{4.1}$$

A plot of the original data is shown in Figure 4.4 and the plot after taking the difference is shown in the Figure 4.5. The last but not the least pre-processing step was the normalisation of data. In normalisation, we scale the values within some specific range usually between 0 and 1 which is called min-max normalisation. Normalisation does not disturb the difference of values or change the data, instead we scale the data to avoid the biases from possible outliers. The formula for min-max is given below:

$$x_n = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.2}$$

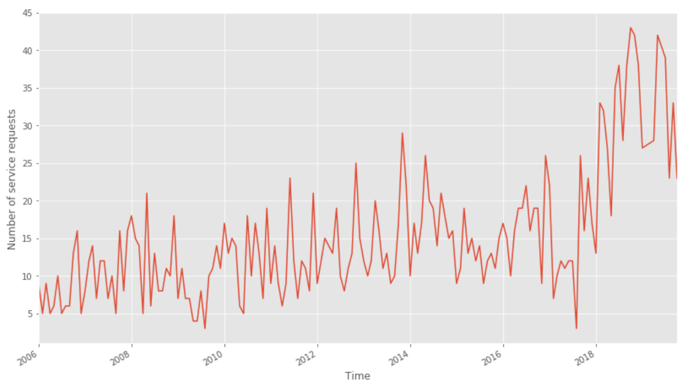

Figure 4.4: Service request frequency per month (Non stationary)

4.1.4 Stationarity

It is important to figure out if the time series is stationary or not (see Section 2.3.6). At the first step, we plotted the data and analysed the data visually. We were able to confirm from the Figure 4.4 that we have some kind of trend in the series plot and might be some seasonality components. We drew the histogram shown in Figure 4.6 to visually inspect the distribution of data. The histogram looks similar to Gaussian (we will not assume Gaussian though) with a long right tale skew due to the few large number of values at the end of series because of general increasing trend. As the data looks much less 'bell-shaped' due to seasonality and trending the data is not stationary.

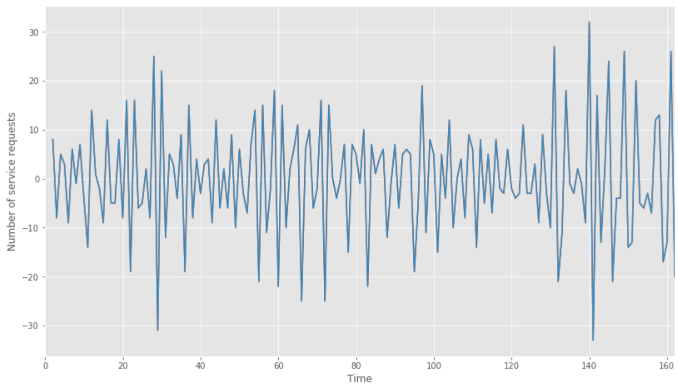

Figure 4.5: Service request frequency per month (Stationary)

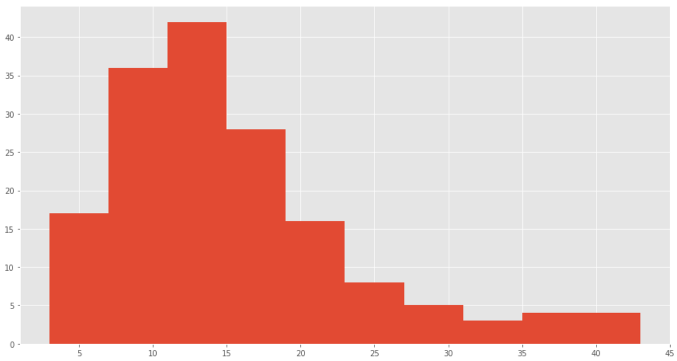

Figure 4.6: Histogram of the series with a positive skewness

As the visual report is not accurate we did some statistical test to confirm our hypothesis about the stationarity of series. We will run two tests using the Python library [19], ADF (Augmented Dickey-Fuller) and KPSS (Kwiatkowski-Phillips-Schmidt-Shin). ADF is an important statistical test to figure out the presence of the unit root in the series data and alternatively help us determine the stationarity in the data. The null hypothesis H_0 is that we have a unit root in the data while there is no unit root is the alternate hypothesis. If we are able to reject the null hypothesis and there is no null root and data is stationary or vice versa. The p-values were taken from regression surface approximation from [9] with updated tables. The results obtained are shown in the table.

The p-value in Table 4.1 is the probability score. Based on p-value we can either reject or accept the hypothesis. If the p-value is less than the

ADF	-1.26
P-values	0.65
Lag	4
Number of observations	158
Critical values 1%	-3.47
Critical values 5%	-2.88
Critical values 5%	-2.58

Table 4.1: ADF test results

predefined alpha level (usually 0.05), we reject the hypothesis. Since the value is 0.65, we are unable to reject the H_0 and it means our data is non stationary. Also the ADF value is not less than the 1% confidence value, so we can safely conclude that the time series is non-stationary.

We run the KPSS test next with the null and alternate hypothesis opposite to the ADF test. The null hypothesis H_0 is that the process is trend stationary. The p-values were interpolated from Table 1 of [15]. As the 4.2 shows that the KPSS test statistics is greater than the critical values and we can reject the null hypothesis H_0 and again conclude that the process is not stationary.

KPSS	0.788
P-values	0.01
Lag	14
Critical values 1%	0.739
Critical values 2.5%	0.574
Critical values 5%	0.463
Critical values 10%	0.347

Table 4.2: KPSS test results

4.2 Experiments

We discussed in Section 3 about the structured time series and ARIMA model for detecting the Trend and seasonality in time series. We implemented both of these time series techniques for our service request data.

4.2.1 Structural Model

The structured and additive model was implemented using the Python library called prophet. The library provides a lot more functions and features for forecasting time series with non linear trends using yearly, weekly, monthly or daily seasonality depending on the use case and data. We used the monthly seasonality setting for our data since we summed the Service Requests per month and assumed the series to have monthly seasonality component.

4.2.1.1 Trend

Prophet allows to use logistic growth model (see Section 3.1.3) for forecasting growth by setting a *cap* value (maximum number that it could reach) depending on the market value and use case. The *cap* is set manually for each data point and the value is usually provided by experts of the industry from which the time series originated. The *cap* value could be set in an increasing order if the data is growing over time. Similarly, a *floor* could be set for the logistic growth model explaining the minimum value that it could take. In our case, the *cap* value was set to be constant value of 70 while the *floor* value was 0. The *cap* and *floor* values were assumed to be the maximum and minimum number of service requests that could be generated in one month in a facility respectively. The logistic growth model did not yield good enough result (Appendix A.1, A.2) because it failed to capture the growth when the number of service requests grew unexpectedly after 2016.

The linear growth model fits very well and was able to capture the trend shown in Figure 4.7 from the data. The model was achieved by selecting the option of automatic change point (see Section 3.1.4) although we can specify the exact location of changepoints by a date in the series depending on when the trend is supposed to change. In our service request use case, there is no such case therefore we selected the automatic changepoint selection option. A range of changepoint prior scale values starting from .4 to .9 with .01 difference were tried one by on as parameter S_j and best one was chosen based on mean squared error. By default, the library chooses the first 80% of the time series to place multiple changepoints. Since we have an abrupt change at the end of series, this limit was increased to 90% to capture the rest of the important change points from the series.

4.2.1.2 Seasonality

The seasonality signal was calculated by a partial Fourier sum (see Section 3.1.5). The number of terms in the partial sum could help to define the parameters that will show how quickly the seasonality will change. We tried

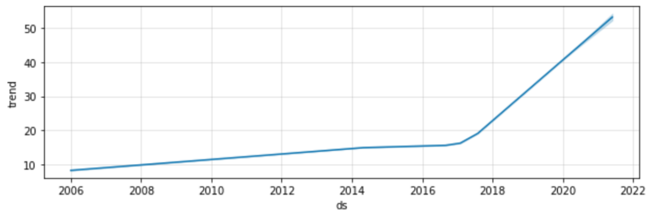

Figure 4.7: Trend with Linear Growth

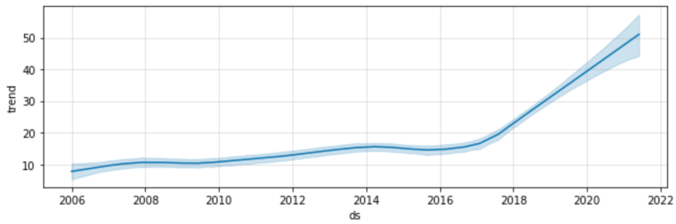

Figure 4.8: Trend with Linear Growth and uncertainty intervals

different values for the number of terms in the Fourier series based on the idea that if we want to capture higher frequency signals we can increase the value. Although increasing the *yearly_seasonality* parameter could lead to over fitting. We were interested in higher level signals in the series and found the parameter value 5 to produce better results without over-fitting.

To get the forecasts, we add the trend and seasonality signals together as an additive model. The library also has the option to add the multiplicative setting. Using either of the options did not have any effects on the seasonality signal alone since its obvious that the additive or multiplicative option affects the final prediction. Based on the prediction accuracy criteria, multiplicative model produced slightly better results than the additive model.

Since we are using a series with monthly data, seasonality was captured shown in Figure 2.3 for the whole year. The graph shown in Figure 2.3 is without any uncertainty measures, there is a possibility to replace MAP estimates with the MCMC sampling to obtain the seasonality with the uncertainty measure.

4.2.1.3 Forecasting

The prophet library returns the predicted y for the actual prediction value along with *yhat* values (upper and lower) as uncertainty interval for the

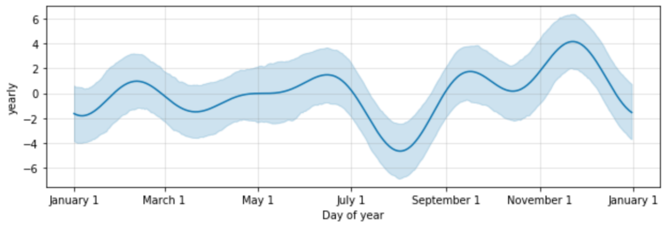

Figure 4.9: Trend with Linear Growth and uncertainty intervals

defined period of time in the future. The uncertainty interval was sourced from future trend changes as the biggest source, seasonality estimates and noise by assuming that the future frequency and trend magnitude will be the same as we have in the history. By projecting these trend changes while moving forward, we obtain the uncertainty distribution. The width of the uncertainty intervals is controlled by the *interval_width* parameter we found .9 value to be the effective interval while trading off with over and under fitting. Even though the model fit seem like very good, future predictions was just a straight line without

The graph below shows the final forecast for the future 20 months based on the seasonality and trend signals. The red lines are the change points that it automatically detected. The dark blue line is the actual prediction signal while the light blue is the uncertainty interval.

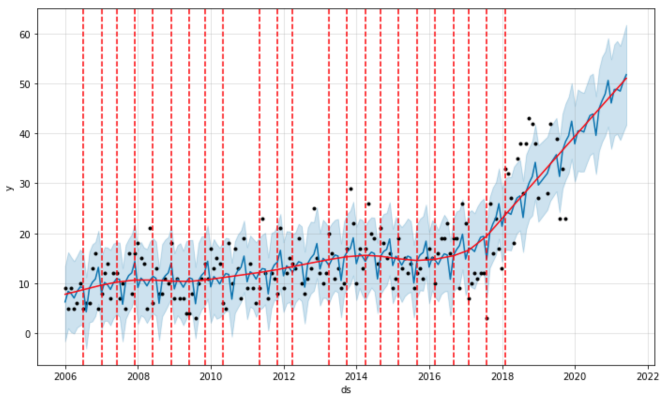

Figure 4.10: Fitting and forecast of service requests

4.2.2 ARIMA Model

Arima (Autoregressive Integrated Moving Average) model was built using its implementation from Python module statsmodels. The implementation fits the model by exact maximum likelihood via Kalman filter [6]. The model takes p, q and d parameters where p is the order of the autoregressive model, q is the order of the moving average model and d is the number of differences required to make the given time series stationary. These parameters were selected by using Grid search over a range of values on both AIC and BIC criterion separately and optimal parameters were chosen on both of these criterion. The model was fitted using the training dataset and predictions were made for the same period. The graph 4.11 shows the actual signal and the model fit. The model was successfully able to ignore the noise signal in the time series. The residual signal captured by the model is shown in Figure 4.12. Even though the model fit is very good, the model prediction is not very plausible and seem to follow a constant trend in the future.

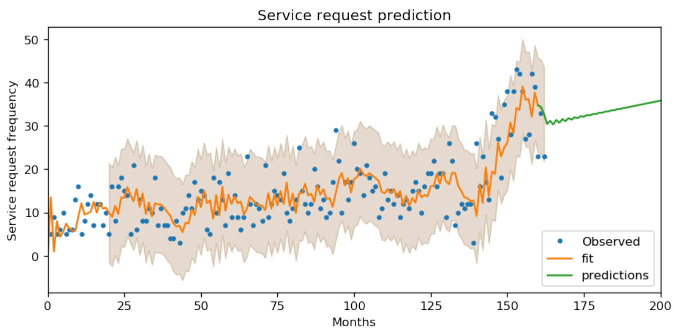

Figure 4.11: Arima model fit and prediction

4.2.2.1 Trend and seasonality

The trend, seasonality and noise signals were decomposed using the seasonal_decompose method of statsmodels module. The method uses the moving averages to decompose the time series signal into trend, seasonal and residual components. We tried multiplicative as well as additive decomposition model and observed the results. The trend component was almost similar in both cases but the seasonal component was different. The Additive model seasonality component shown in Figure 4.13 was comparable to the one we created using the structural model shown in Figure 4.8. The

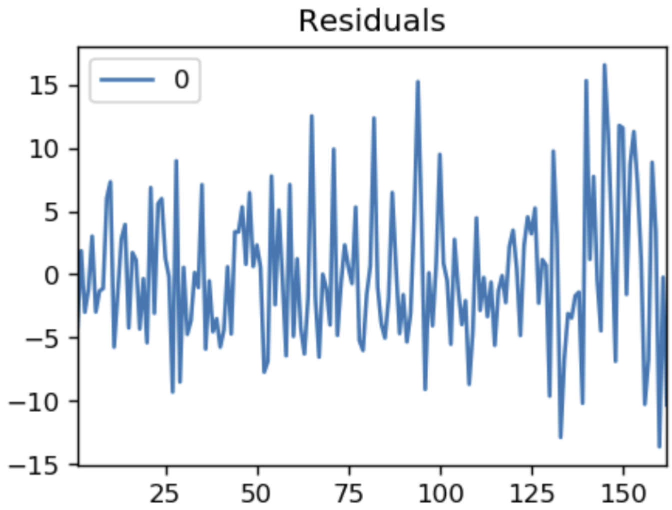

Figure 4.12: Residual signal captured by Arima model

seasonal component is repeatedly shown for many years, one repeated cycle corresponds to one year cycle.

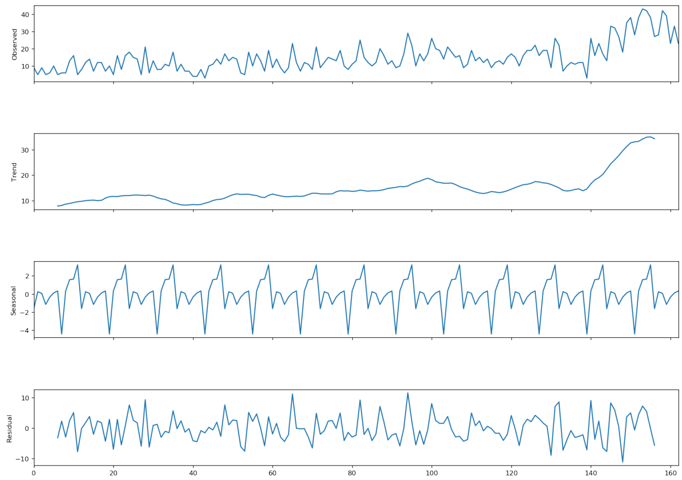

Figure 4.13: Signal decomposition using additive model

Chapter 5

Evaluation

It is important to define the prediction objective and track the performance of the model. Additionally, it is crucial to measure how error-prone the model is so that the business important decisions could rely on the forecasts made by the model.

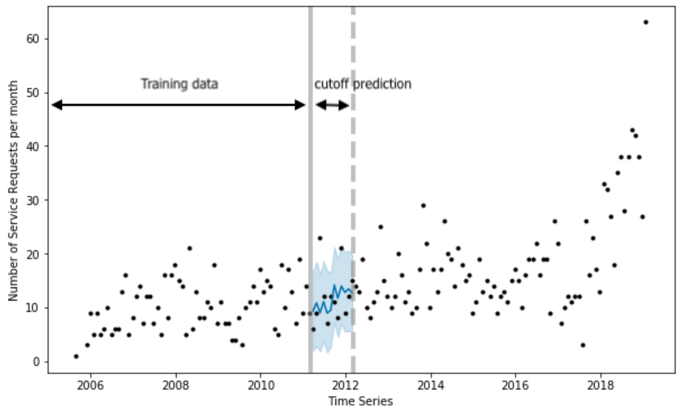

Figure 5.1: Prediction for the cutoff region

5.1 Structural model evaluation

The structural model was implemented using the Python library Prophet, the model was evaluated using the Prophet functionality to measure the prediction error using the historic data which were used to train the model. The functionality chooses multiple cutoff points from the data and the model is fitted until that particular cutoff point only and this is repeated for all the selected cutoffs. The actual and the predicted values are then compared to calculate the errors. We specified initial size of the training set, forecast horizon which is the number specifying the time we want to predict and the size of cutoffs. The forecast was made for the cutoffs and the horizon and error is computed by using the true value y and out-of-sample forecast value $y^{'}$. The process could be defined as $y^{'}(t|T)$, where $y^{'}$ is the prediction made for time t using the T historical data. While $d(y, y^{'})$ is a distance metric such as mean absolute percentage error (MAPE). The Figure 5.1 shows a prediction made using the cutoff. The data was used only until the cutoff region in the graph to the train the model and the 15 predictions were made for the selected period from the rest of the series.

The table 5.2 shows the different accuracy measures used to find the difference between predictions and the actual values. It can be noticed from the table that the prediction made for fewer days have huge MSE error, while it seems to reduce with increasing horizon. The graph

	horizon	mse	rmse	mae	mape	coverage
0	22 days	10.892073	3.300314	3.300314	0.550052	1.0
1	53 days	3.372813	1.836522	1.836522	0.204058	1.0
2	83 days	196.100569	14.003591	14.003591	0.608852	0.0
3	114 days	0.836263	0.914474	0.914474	0.076206	1.0
4	144 days	3.737766	1.933330	1.933330	0.276190	1.0

Figure 5.2: Accuracy of predictions made for the cutoffs

5.2 Arima model evaluation

It was clear by inspecting and comparing the predictions visually from the Arima model shown in Figure 4.11 with the actual values that the predictions were following a slow constant trend which did not correspond to the historic or the future data of series and thus were unreliable. Although, the trend and especially the seasonal component obtained from decomposition of the signals using Arima model was quite similar to the one obtained from structural time series model. It is important to mention that we could not use the methods like cross validation since the observations were not exchangeable and the series was non stationery also. The time series showed a sudden increase in the number of service request at the end of series and this trend was not observed in the past data, because of this reason, it was difficult to split the data into test and training sets. The model could not learn the trend unseen in the historic data. Although the sudden increase in the trend was according to the annual seasonal component.

The model selection for Arima was done using the AIC and BIC criterion [7]. Both AIC and BIC are penalized-likelihood information criterion, AIC estimates the difference between the unknown true likelihood function and the one fitted by the model for the given data while BIC estimates the function of posterior probability of the model being true.

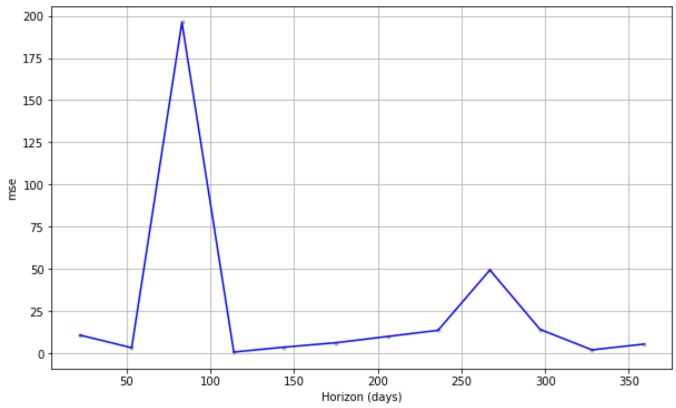

Figure 5.3: Change in MSE for different prediction periods

These parameters were selected by using Grid search over a range of values on both AIC and BIC criterion separately and optimal parameters were chosen on both of these criterion. The table 5.1 shows the different models with corresponding AIC and BIC values out of which the best model was chosen with the lowest AIC and BIC values. The model was fitted using the training dataset and predictions were made for the same period. The graph 4.11 shows the actual signal and the model fit.

AIC	BIC	MSE
1150.95	1157.12	12.26
1101.60	1110.86	9.93
1087.55	1093.71	8.09
1021.46	1030.71	7.28
1066.72	1075.98	8.48
1058.01	1067.26	7.57
1023.36	1035.69	7.35
1053.64	1065.99	7.95
1038.48	1050.81	7.02
1022.79	**1038.20**	**7.32**

Table 5.1: Model comparison with different AIC, BIC and RMSE scores

Chapter 6

Discussion

About 43% of the activities in non housing sector are related to maintenance [21]. Also, a strong link between maintenance and company is shown by many authors in terms of profitability and how competitive the company is [18][2][13] [12]. It is thus important to do the maintenance not only at the right cost but also at the right time. We took the data of a commercial facility and tried to predict the future number of service requests in one month and analyzed the past trend in more than 10 years of monthly data. Our goal was to enable the stakeholder to make better decision at right time, reduce the maintenance cost and improve the customer satisfaction.

We were able to get the overall trend (shown in Figure 4.7) of number of service requests over the years and a gradual increase was noticed. This could be very helpful information for the managers of the facility to understand how much they are spending on maintenance and how it will continue to grow over time and at what point in the future it would be too much to sustain it. This could also be useful when a real estate firm is going to lease or purchase new facilities. It is important to understand that this trend might have other explanations; the number of tenants might have increased over the time resulting in higher maintenance need or the building components and equipment might have worn out and require more than needed maintenance. We leave these explanations to be figured out by the domain experts and draw conclusions from these results. The seasonality was another important aspect of the results helping with maintenance resource planning. The seasonality provided the increase and decrease in number of service request observed every year over the years (see Figure 4.9). The seasonality graph provided insights into what time of the year is the busiest in terms of maintenance. This enabled the service providers and facility managers to plan the maintenance for these busiest months to provide better service and increase the customer satisfaction. This could also be helpful in reducing the maintenance cost for these months. It is interesting to notice that the seasonality patterns are according to the social aspects of people in Finland. Least number of service requests were created during summer holiday season when most people were away from the facility. Again this interpretation could be wrong and should be concluded by domain experts while keeping in mind other parameters and aspects of the facility.

Finally the predictions were supposed to be used by the decision makers of the facility to have an idea about what to expect in the future and plan for it beforehand. The results included the uncertainty measure also to give a general idea about the future number of service requests rather than providing a point prediction. The uncertainty measure was especially important because it is necessarily not important to forecast the exact number of service requests but to get a range of values so that the reliability on the forecasts can be managed.

The model was trained for each facility's data separately and will be deployed the same way. Each real estate have different number of tenants and capacity, service company could be different for diffident facilities, the age, size and location of the facilities vary and the availability of the data was different due to missing data points.

Chapter 7

Conclusions

In this study, we were trying to analyze and predict the real estate service request data and the goal was to be able to fit a model that would provide some insights into the historic data and maybe predict the future values. The results, in terms of seasonality and trend, looks promising. The time series data was univariate, taking into account just the number of service requests. The predictions could be improved further and model could be made more robust by adding more data and possibly other indicators like the number of people using the facility, area of the facility, responsible service company, building age etc.

While comparing the used methods for time series analysis; structural time series model and arima model, based on predictions, the structural time series seem to perform better. Although there was no significant difference between seasonality and trend in both models. Arima trend component seem to be little bit over-fitting and seem to adapt abruptly to every rise and fall while the structural time series trend was smooth. The predictions from Arima model was suffering from the mean convergence issue in long-term time series forecasting [23].

The sudden rise in the number of service requests at the end of the series made it difficult to slice the data at the end to make predictions for that period as a testing step. It was considered obvious that the model would not be able to predict the increase because it was not noticed in the past data. Since the data points were considered to be non exchangeable, we could not split the data from the whole series randomly for cross validation or testing purposes also. This could be improved further in future work by adding more data to the tail of the series.

Bibliography

[1] Environmental limits to population growth. https://cnx.org/contents/GFy_h8cu@11.10:eeuvGg4a@4/Environmental-Limits-to-Population-Growth/, 2020.

[2] Al-Najjar, B. The lack of maintenance and not maintenance which costs: A model to describe and quantify the impact of vibration-based maintenance on company's business. *International Journal of Production Economics 107*, 1 (2007), 260–273.

[3] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time Series Analysis: Forecasting And Control.* John Wiley & Sons, 2015.

[4] Brockwell, P. J., and Davis, R. A. *Introduction to Time Series and Forecasting.* springer, 2016.

[5] Chatfield, C. *Time-Series Forecasting.* Chapman and Hall/CRC, 2000.

[6] Diebold, F. X. Exact maximum-likelihood estimation of autoregressive models via the kalman filter. *Economics Letters 22*, 2-3 (1986), 197–201.

[7] Dziak, J. J., Coffman, D. L., Lanza, S. T., Li, R., and Jermiin, L. S. Sensitivity and specificity of information criteria. *bioRxiv* (2019), 449751.

[8] Farmer, J. D., and Sidorowich, J. J. Predicting chaotic time series. *Physical review letters 59*, 8 (1987), 845.

[9] Hamilton, J. D. *Time Series Analysis*, vol. 2. Princeton university press Princeton, NJ, 1994.

[10] Harvey, A., and Peters, S. D. Estimation procedures for structural time series models.

[11] Harvey, A. C., and Shephard, N. 10 structural time series models.

[12] Ingwald, A. *Technologies For Better Utilisation of Production Process Resources*. PhD thesis, Institutionen för teknik och design, 2009.

[13] Kans, M. *On the Utilisation of Information Technology for the Management of Profitable Maintenance*. PhD thesis, Institutionen för teknik och design, 2008.

[14] Klose, C., Pircher, M., and Sharma, S. Univariate time series forecasting. *Term Paper* (2004), 1–46.

[15] Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics 54*, 1-3 (1992), 159–178.

[16] Nie, S.-y., and Wu, X.-q. A historical study about the developing process of the classical linear time series models. In *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013* (2013), Springer, pp. 425–433.

[17] Perret, J., and Jouvent, M. *Guide de la maintenance des bâtiments: diagnostic d'un patrimoine bâti existant, prévention des désordres et actions pour y remédier 308 fiches tecniques de suivi des ouvrages*. Le Moniteur, 1995.

[18] Pulselli, R., Simoncini, E., Pulselli, F., and Bastianoni, S. Emergy snalysis of building manufacturing, maintenance and use: Embuilding indices to evaluate housing sustainability. *Energy and buildings 39*, 5 (2007), 620–628.

[19] Seabold, S., and Perktold, J. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference* (2010).

[20] Shumway, R. H., and Stoffer, D. S. *Time Series Analysis and Its Applications: With R Examples*. Springer, 2017.

[21] Silva, J., and Falorca, J. A Model Plan for Buildings Maintenance With Application in the Performance Analysis of a Composite Facade Cover. *Construction and Building Materials 23*, 10 (2009), 3248–3257.

[22] Slutzky, E. The summation of random causes as the source of cyclic processes. *Econometrica: Journal of the Econometric Society* (1937), 105–146.

[23] Swastanto, B. A. Gaussian process regression for long-term time series forecasting.

[24] Veale, P. R. Managing Corporate Real Estate Assets: Current Executive Attitudes and Prospects for an Emergent Management Discipline. *Journal of Real Estate Research 4*, 3 (1989), 1–22.

[25] Walker, G. T. On periodicity in series of related terms. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 131*, 818 (1931), 518–532.

[26] Wikipedia. Fourier Series — Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/w/index.php?title=Fourier%20series&oldid=925623685`, 2019. [Online; accessed 14-November-2019].

[27] Wold, H. *A Study in the Analysis of Stationary Time Series*. PhD thesis, Almqvist & Wiksell, 1938.

[28] Yule, G. U. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 226*, 636-646 (1927), 267–298.

Appendix A

Appendix

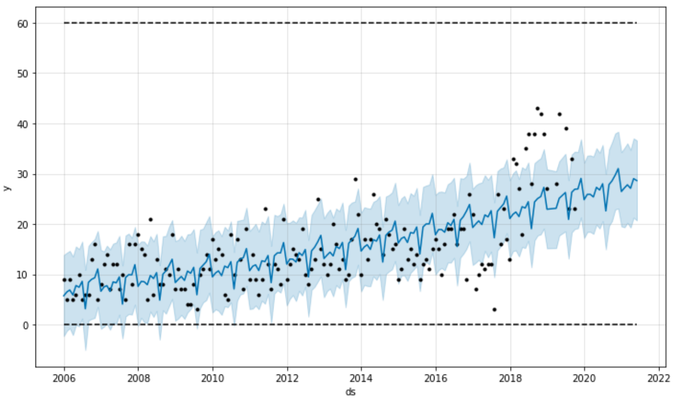

Figure A.1: Fitting a Logistic Growth model

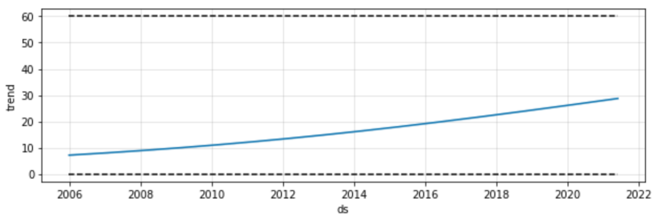

Figure A.2: Trend with Logistic Growth

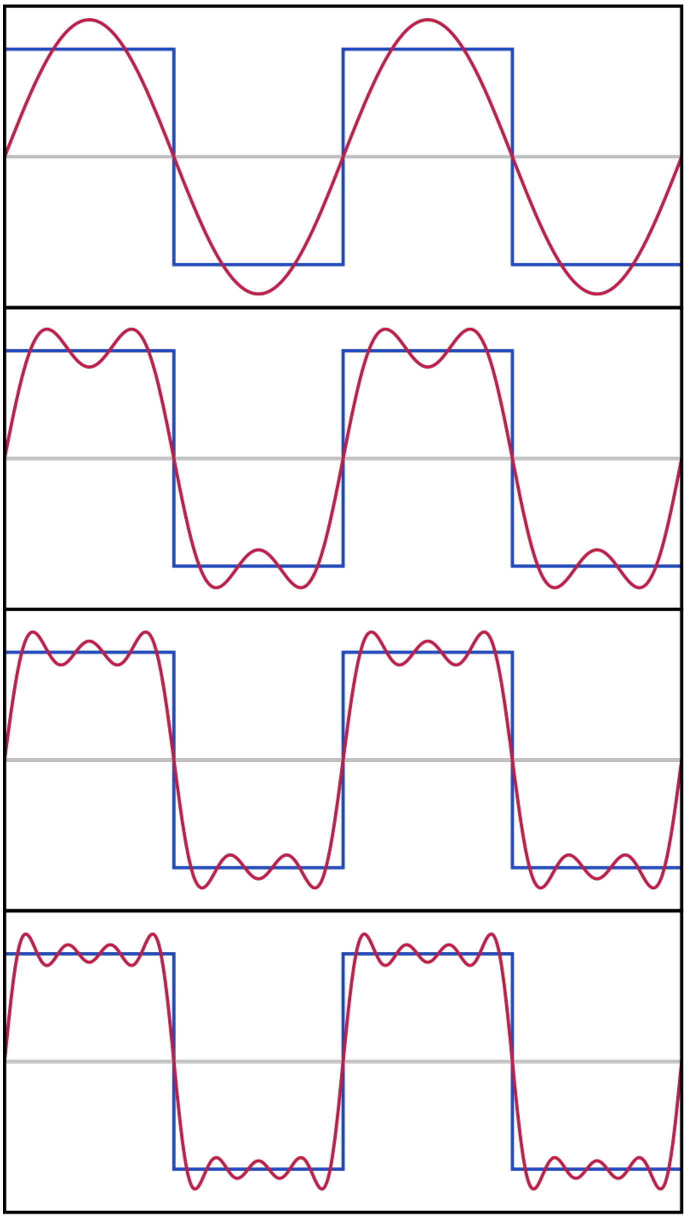

Figure A.3: Partial sum of first four Fourier Series [26]